APPLYING MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING TECHNIQUES TO TWITTER SENTIMENT CLASSIFICATION FOR TURKISH AND ENGLISH

by

Değer Ayata

M.S., Computer and Control Engineering, Istanbul Technical University, 2000

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2018

ACKNOWLEDGEMENTS

I would like to thank to my supervisors Prof. Murat Saraçlar and Assoc. Prof. Arzucan Özgür for their patience, guidance and valuable feedbacks. The completion of this thesis could not have been possible without their understanding and consultancy. I would also like to thank to jury members Prof. Levent Arslan and Prof. Banu Diri for kindly accepting to be in my thesis jury and providing valuable feedback.

I would like to thank to friends for their support and encouragement. Şeniz Gayde inspired me to start to natural language processing domain and supported me by sharing her valuable ideas. John Femi Abe helped me to fix some latex problems. Deep Learning readings and discussions at Bogazici Computer Department with Mine Öğretir, Arda Çelebi, Onur Güngör and Çağıl Uluşahin were always helpful and amazing.

I would like to dedicate this thesis to my parents Nehiye and Bedri Ayata. My mother was the first person I have informed when I have been accepted to Boğazici Electrical and Electronics Engineering Master of Science program.

I would like to express my special thanks to my parents, brothers Hakan and Mesut, my sisters Güllü, Fatma, Selda and Duygu, for their love, trust, understanding and every kind of support not only throughout my thesis but also throughout my life. Güllü helped me in data annotation, Mesut, Selda and Duygu read all my conference papers and helped me to publish better papers, Hakan and Fatma motivated me to continue and finalize this thesis.

ABSTRACT

APPLYING MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING TECHNIQUES TO TWITTER SENTIMENT CLASSIFICATION FOR TURKISH AND ENGLISH

Determining the feelings of a user on a topic is called sentiment analysis. Sentiment analysis is done so as not just to get a sort of feedback data for decision making but also to check the emotions, characteristics and influence of situation actions, news or events on users or specific data with respect to users. Sentiment analysis of a sentence is the effort of understanding whether the sentence contains positive, negative or neutral meaning and is used to gather overall attitude toward a product, person or event etc. In this thesis we have studied the effectiveness of applying machine learning and statistical natural language processing techniques collaboratively to Twitter messages based sentiment classification problem. This thesis covers the work on Turkish tweet sentiment analysis, sector based sentiment analysis, English tweet sentiment analysis, and political opinion prediction. We introduce sector based sentiment analysis framework by applying machine learning and statistical natural language techniques collaboratively. We apply our framework to finance, telecom, retail and sport sectors. In English sentiment analysis, this thesis covers our solution for the SemEval Twitter Sentiment Analysis task. Consecutively, in English Twitter analysis our work also addresses the task of political orientation classification based on Twitter data. We have used various machine learning algorithms that can predict and automatically classify whether a tweet belongs to a republican or a democrat. We have used a Twitter dataset which consists of democrat and republican voters tweets. This work covers political opinion/sentiment tendency estimation based on Twitter messages of voters.

ÖZET

TÜRKÇE VE İNGİLİZCE TWİTTER DUYGU SINIFLANDIRMASI İÇİN MAKİNE ÖĞRENMESİ VE DOĞAL DİL İŞLEME TEKNİKLERİNİN UYGULANMASI

İnsanların belirli bir konu hakkındaki duygularını anlama faaliyetleri duygu analizi ya da duygu çözümlemesi olarak tanımlanmaktadır. Duygu çözümlemesi sadece karar süreçlerinde geri bildirim almak için değil, kullanıcıların olaylar karşısındaki duyguları, haber ve olayların etkisini belirlemek için de yapılmaktadır. Cümle bazında duygu çözümlemesi ise kullanıcıların bir ürün, kişi vb. hakkında yazdığı mesajların olumlu, olumsuz veya nötr sınıflarından hangisini taşıdığını belirleme işlemidir. Bu tezde Twitter verisi için duygu çözümlemesi probleminin çözümünde makine öğrenmesi ve istatiksel doğal dil işleme tekniklerinin beraber kullanımının verimliliği çalışılmıştır. Tez Türkçe tweet duygu çözümlemesi, sektöre dayalı duygu çözümlemesi, İngilizce tweet duygu çözümlemesi ve politik yönelim tahminlemesi başlıklarında yaptığımız çalışmaları kapsamaktadır. Sektöre özel Türkçe tweet duygu çözümlemesi için makine öğrenmesi ve istatiksel doğal dil işleme tekniklerinin birlikte kullanımı ile bir çerçeve önerilmiş ve önerilen çerçeve finans, perakende, telekomünikasyon ve spor sektörlerine uygulanmıştır. İngilizce duygu çözümlemesinde, bu tez SemEval-2017 Twitter duygu çözümlemesi'nde ele aldığımız yöntem ve çalışmaları kapsamaktadır. Buna ek olarak, İngilizce temelli çalışmada duygu çözümlemesinde kullandığımız yöntemler ile seçmen tweet iletilerinden oluşan bir veri kümesi kullanılarak Twitter mesajları analiz edilerek politik yönelim analizi çalışılmıştır. Politik yönelim analizinde seçmenlerin Twitter mesajlarına göre eğilimlerinin 'demokrat' veya 'cumhuriyetçi' sınıflarından hangisine ait olduğu tahminlenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c^*	predicted class
h_{NB}	Naive Bayes function
n	Tweet word count
N	Total number of tweets
T_i	Tweet i
Twt	Tweet
$twtVec$	Tweet vector
w	word
$wdVec$	word vector

LIST OF ACRONYMS/ABBREVIATIONS

BoW	Bag of Words
CCR	Correctly Classification Rate
DF	Document Frequency
DT	Decision Tree
IG	Information Gain
IDF	Inverse Document Frequency
LSTM	Long Short Term Memory Recurrent Neural Networks
mRMR	minimum Redundancy Maximum Relevance
ML	Machine Learning
NLP	Natural Language Processing
sNLP	Statistical Natural Language Processing
NB	Naive Bayes
RF	Random Forest
RNNs	Recurrent Neural Nets
SVM	Support Vector Machine
TF	Term Frequency

1. INTRODUCTION

Determining the feelings of a user depending on a topic is called sentiment analysis (SA). Sentiment Analysis is done so as not just to get a sort of feedback data for decision making but also to check the emotions, characteristics and influence of situation actions, news or events on users or specific data with respect to users. SA is also known as sentiment mining, appraisal detection, subjectivity extraction, mood analysis, opinion mining or feeling detection. SA is studied as part of affective computing which contains work and tasks on human - computer interaction and recognition of human expressions [1, 2, 3]. SA is studied via sentiment units, which can be small units [4], paragraphs or whole [5] documents based on the needs and problem formulation. Sentiment analysis of a sentence is the effort of understanding whether the sentence contains positive, negative or neutral meaning and is used to gather overall attitude toward product, person or event etc. Another perspective of SA is to extract personal, abstract, biased data from human beings via intelligent automated processes by using a rich set of tools including lexicons, rules, computational linguistic analysis, statistical natural language processing (NLP), pattern recognition and machine learning (ML).

Twitter is useful for learning recent news both about acquaintances and interested topics. Twitter is being used by many Internet users. Twitter is among the most popular social platforms in which users generate micro-blogs (tweets) within a limited character space. The messages are short and informal and detecting sentiment information automatically is a non deterministic and hard task. Social media reviews and especially Twitter messages (tweets) are often less than 140 characters and not structured. Tweet based analysis methods usually have challenges including dynamic and evolving jargon, spelling errors, non - grammatical language usage, dynamic abbreviations, emoticons. These challenges make SA a complex problem. Existing knowledge base, lexicons and rule based approaches do not work well in this dynamic and volatile environment. Such methods usually can not capture vital data for SA. Analyzing the data in micro-blog text is important for different purposes including advertising, personalized services etc. Banks, telecom operators and big enterprises have customer feedback centers and are investing on customer services to understand their customers need better and to shape their strategies. Customers of big enterprises are using internet and parallel to internet usage micro-blogging and text messaging is increasing tremendously.

The information on the Internet cannot be tracked manually. Automated sentiment analysis is crucial for company customer services and gives them capability to capture complaints and/or positive feedbacks in the right time. Implementing an efficient sentiment analysis tool will increase customer satisfaction, decrease operational costs, decrease complaint resolution times, enable implementing customer review based new products, enable to track sentiment score. SA enables detecting negative feedbacks and raises real-time alerts so that quick actions can be taken. As a vital part of social listening SA helps detect angry and unhappy customers.

Subjectiveness and SA focus on the automatic detection of private states such as tendencies, emotions, and sentiments. To date, a large number of NLP applications have used methods for automatic SA including tracking sentiment timelines in social media forums, mining opinion from product reviews, automatic expressive text-to-speech synthesis or as a first-phase filtering in question answering systems and text semantic analysis [6].

1.1. Purpose and Motivation

Although there are a lot of studies for English language, Turkish language based sentiment analysis studies are limited and this problem has not been solved for Turkish language at the moment. In this thesis we have studied the effectiveness of applying machine learning and statistical natural language processing techniques collaboratively to twitter messages based sentiment classification problem both for Turkish and English. The thesis covers the work on Turkish tweet sentiment analysis, sector based sentiment analysis, English tweet sentiment analysis, as well as political opinion prediction.

We have aimed to answer sentiment analysis related questions within this thesis work. The driver questions of this thesis are as follows:

(i) Is it possible to apply Machine Learning and Natural Language Processing techniques collaboratively for Turkish and English?
(ii) Do sector based sentiment analysis approaches work better than general sentiment analysis?
(iii) Do word vector representations work better than bag of words?
(iv) Do sequential methods better than non-sequential methods?
(v) Does sentiment analysis pipeline work for political sentiment analysis?

1.2. Contribution of the Thesis

In this thesis, the effectiveness of applying ML and statistical NLP techniques collaboratively to twitter messages based sentiment classification problem has been studied. This thesis covers the work on Turkish tweet sentiment analysis, sector based sentiment analysis, English tweet sentiment analysis, and political opinion prediction.

In Turkish tweet sentiment analysis task we have studied on processing and categorization of Turkish tweets. A framework for supervised SA of Twitter Data in Turkish has been proposed. The results for sentiment analysis on Twitter with ML and NLP methods are presented.

Sector based sentiment analysis framework by applying machine learning and statistical natural language techniques collaboratively has been introduced. The framework has been applied to finance, telecom, football and retail sectors. By using Support Vector Model (SVM), Random Forest (RF), Naive Bayes (NB) and word embedding models, tweets pertaining to various sectors have been vectorized, classified and results have been compared. It has been shown that, sector based tweet classification is more successful compared to general tweet classification approaches for some sectors.

In English based sentiment analysis our work contains two tasks. The first task explains our solution for SemEval-2017 Task four: SA in Twitter. We have participated in Semantic Evaluation Subtask A: Message Polarity Classification subtask and implemented two different systems. The first system is based on word embeddings, uses word embeddings for feature representation and vectorization, and SVM, RF, NB algorithms for the classification of tweets into one of positive, negative and neutral classes. The second system uses Long Short Term Memory (LSTM) which is a specialized sequential Recurrent Neural Networks classifier and uses word indexes as sequence of inputs.

Consecutively, the second focuses on tweet based political orientation prediction. We have used various machine learning algorithms that can predict and automatically classify whether a tweet belongs to republican or a democrat. We have used twitter dataset that was produced by democrat and republican voters. This work covers political opinion/sentiment tendency estimation based on Twitter messages of voters.

During our thesis work we have produced papers and participated in relevant conferences. In total four conference papers have been produced and accepted to national and international conferences. Our aim was to collaborate, share and discuss our findings immediately. Below there is a list of produced papers:

- Turkish tweet sentiment analysis with word embedding and machine learning [7].
- Sentiment Analysis with Word Embedding and Long Short Term Memory RNN Approaches [8].
- Political opinion/sentiment prediction via long short term memory recurrent neural networks on Twitter [9].
- Sector based Sentiment Analysis Framework for Social Media via Machine Learning and Natural Language Processing Approaches [10].

1.3. Organization of the Thesis

This thesis document contains Abstract, Introduction, Background and Related Work, Methods and System Architecture, Sentiment Analysis in English, Sentiment Analysis in Turkish, Conclusion chapters and References respectively.

Background and Related Work chapter provides the previous studies and pertaining work for sentiment analysis including natural language processing methods, machine learning algorithms and literature review on related work. The details about used classification methods,important studies both in English and Turkish sentiment analysis are presented in this chapter.

The rest contains information about our work and application in this thesis. Methods and System Architecture chapter contains our common architectures developed for both English and Turkish sentiment analysis and political opinion prediction. Sentiment Analysis in English chapter continues with our SemEval Sentiment Analysis Task and twitter based political opinion/sentiment prediction studies which address the task of political orientation prediction. Sentiment Analysis in Turkish chapter introduces our sector based sentiment analysis framework which is proposed to predict sentiment polarity in different sectors.

2. BACKGROUND AND RELATED WORK

This chapter provides previously done studies and pertaining work for sentiment analysis including natural language processing methods, machine learning algorithms and literature review on related work. The details about used classification methods, used datasets, important studies both in English and Turkish sentiment analysis are presented.

2.1. Sentiment Analysis

There is a large amount of study in SA, especially in the domain of news, movie reviews, blogs, complaint management, customer feedback and product reviews. SA focuses on the automatic detection of human expressions and states such as tendencies and opinions. To date, a large number of NLP applications have used techniques for automatic SA including tracking sentiment timelines in social media forums, mining opinion from product reviews, automatic expressive text-to-speech synthesis or as a first-phase filtering in question answering systems and text semantic analysis [6].

One of the earliest studies for Twitter is presented by Rao *et al.* [11]. Rao *et al.* aimed to find latent attributes of users, the social network structure and communication behavior in Twitter [11]. Rao *et al.* tried to detect gender, age, regional origin, and political orientation through tweets. A few algorithms focus on capturing sentiment strength as well as sentiment polarity [3, 12, 13, 14, 15, 16]. Studies on detecting sentiment polarity and sentiment strength work on the basis that humans can differentiate between mild and strong emotions in text. For example, hate can be defined as a stronger negative emotion than dislike. The main goal of sentiment strength algorithms is to assign a numerical value to texts in order to indicate the strength of any detected sentiment.

In standard ML approaches annotation and labeled data are vital [17]. Generally polarity and strength labeling is done manually by humans. Labeled or annotated data is used to train a model. Sometimes whole documents are studied as a sentiment unit [5], but it's generally agreed that sentiment resides in smaller linguistic units [4]. The text features used are typically sets of most frequently used words, word pairs, word triples, and POS tags in the texts. The features are used to train classification algorithms in order to predict text polarities [3, 4, 18, 19]. Gul *et al.* [3] researched the performance of various machine learning techniques including Naive Bayes, Maximum Entropy and Support Vector Machines. In their work they considered the problem of classifying documents not by topic, but by overall sentiment. Kokciyan *et al.* studied sentiment Classification in Twitter using rich feature sets [20]. In their study, two systems, one for task 1 (Maximum Entropy Model) and one for task 2 (Maximum Entropy and Naive Bayes Model) have been used and results showed that using rich feature sets with machine learning algorithms is a promising approach for sentiment classification in Twitter.

SemEval (Semantic Evaluation), is an ongoing series of evaluations of computational semantic analysis systems [21]. SemEval has sentiment analysis tasks starting from year 2007. In Semeval, there are many sub-tasks such as expression-level task (Contextual Polarity Disambiguation) and a message-level task (Message Polarity Classification). In Contextual Polarity Disambugation task the aim is (given a message containing a marked instance of a word or phrase), determining whether that instance is positive, negative or neutral in that context whereas in Message Polarity Classification task the goal is given message, classifying whether the message is of positive, negative or neutral sentiment.

There are many studies on sentiment analysis focused on English language whereas there are only limited number of sentiment analysis studies for Turkish language. In "Sentiment Analysis in Turkish" titled thesis, Umut Erogul introduced two Turkish datasets tagged with sentiment information and applied existing methods for English [22]. Turksent is a SA and labeling tool developed for social media posts in Turkish language [23, 24]. Another Turkish language specific study is done to compare text representation methods for Turkish Amasyalı *et al.* [13]. Vural *et al.* [25] customized Sentistrength sentiment analysis library by translating its lexicon to Turkish and applied the framework to classify polarity of movie reviews, they used a large corpus of Turkish movie reviews and they stated that although the framework is unsupervised, the performance approached to the performance of supervised polarity classification. Kaya [18] studied SA of political columns by considering transfer learning approaches in Turkish. In transfer learning the aim is to extract needed knowledge from one or more tasks and then to transfer extracted information into a target task. Atan [26] studied the relations between financial status of Borsa Istanbul enterprises and published financial news about them. Amanet [27] studied on Twitter data using the emotion categories like “Happy”, “Appreciation” etc. and most effective words for each emotion are determined. Turkmen [28] studied aspect based sentiment analysis method which is capable of classifying Turkish customer reviews on Internet as positive or negative. Kamburoglu [29] studied predicting Turksh movie review scores using adjective clustering.

2.2. Machine Learning

Machine learning based studies can be categorized into supervised, unsupervised and semi-supervised topics. Feature engineering and feature selection are also vital in a machine learning pipeline.

Supervised learning is one of the most used approaches in ML domain. Although supervised learning is successful in rich set of applications, it has many challenges. The biggest challenge is text records or examples should be annotated manually in order to construct a supervised ML pipeline. In our case, each tweet should be labeled manually by a human coder. Although crowd sourcing like approaches can be used to increase annotation speed, when the number of records increase drastically, annotation will be slow and an expensive method [17]. Another dimension of supervised learning is the need of handcrafted features. SA text attributes (features) are typically sets of word frequencies, term frequencies, words, word pairs and word triples [3, 4, 18, 19].

Unsupervised learning based methods are prefered for many reasons: The first reason is the wide range and variety of products and services being reviewed, the framework must be robust and easily transferable between domains. The second reason is due to the nature of the data. Social media reviews and especially Twitter messages(tweets) are often short and unstructured, and may contain spelling, grammatical errors, slang or specialized jargon. Unsupervised learning methods, are not influenced by the lexical form, and can handle unknown words or word-forms, provided they occur frequently enough [30].

Graph-based algorithms are being used to identify word similarity and relatedness [12]. During the preprocessing of the data, similar word evaluation can be done. A similarity metric to identify similarities between words can be used. There are different similarity metrics like Jaro-Winkler distance metric, Jaccard distance, Euclidian distance, Levenstein distance etc. After calculating similarities between words a similarity graph can be generated where nodes represent words and weighted edges represent similarity values between words. A graph - based method that has been used effectively for sentence-level subjectivity analysis is min-cut algorithm that generates subjective extracts of movie reviews, which in turn can be used to improve methods for sentiment analysis [31]. Pang *et al.* [31] study consist of different steps. In the first step, a graph is constructed by adding all of the sentences in a review as nodes and by drawing edges based on sentence proximity. Then, in the second step, each node is initially assigned a score indicating the likelihood of the correspondence sentence being objective or subjective, based on estimate provided by a supervised subjectivity classifier trained on manually annotated data. Then, in the third step, a min cut algorithm is applied to the graph and used to separate subjective and objective sentences from each other [12].

A classic paper by Peter Turney [32] explains a method to do unsupervised sentiment analysis (positive/negative classification) using only the words *excellent* and *poor* as a seed set. Turney presented a simple unsupervised learning algorithm for classifying a review as “recommended” or “not recommended” [32]. Review classification is done by predicting the average semantic orientation of the phrases in the review that contain adjectives or adverbs. Turney’s approach consists of three steps. The first step is to use a part-of-speech tagger to identify phrases in the input text that contain adjectives or adverbs. The second step is to predict semantic orientation of each extracted phrase. A phrase has a positive semantic orientation when it has good associations and a negative semantic orientation when it has bad associations. Turney calculated the semantic orientation of a phrase as the mutual information between the given phrase and the word “excellent” minus the mutual information between the given phrase and the word “poor”. The third step is to assign the class. A review is classified as recommended if the average semantic orientation of its phrases is positive. Otherwise, the predictions that the item is not recommended [32].

2.2.1. Feature Selection

Feature Selection (FS) considers feature (attribute) contribution to classification. Information Gain (IG) and mRMR (minimum Redundancy Maximum Relevance) are widely used feature selection/reduction algortihms in text processing.

IG is a good feature selection algorithm for text classification. The idea behind IG is to select features that reveal the most information about classes. Such features are highly discriminative.

mRMR is a feature selection algorithm. The central assumption when using a feature selection technique is that the data contains many redundant or irrelevant features. In mRMR, the goal is to select a feature subset that best characterizes the statistical property of target classification variable. mRMR feature-selection method can use either mutual information, correlation, *distance/similarity* scores to select features [14, 33] .

2.2.2. Classification Algorithms

In this thesis study we have used Long Short Term Memory, Decision Trees, Random Forests, Naive Bayes, Support Vector Machine as machine learning classification algorithms. A classifier is vital part for assigning a tweet class category after preprocessing steps.

Long Short Term Memory (LSTM) networks have been proposed to avoid the long-term dependency problem in Recurrent Neural Networks [34]. Although LSTM architecture is very close to RNNs, LSTMs use different functions. LSTM architecture has a chain like form and composed of cells. Memory units are called cells in LSTM architecture. Cells are processing units where the previous state, current input and output is processed and an output is produced.

LSTM networks use state information and for every input LSTM decide which data is going to be throw away and what data is going to be stored. This throw away mechanism is handled by a sigmoid layer called the “forget gate”. It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1} as shown in equation 2.1.

$$f_t = \sigma(W_f[h_{t-1}, x_t) + b_f) \tag{2.1}$$

Decision Trees (DT) are non-parametric ML models that learns rules from data. DTs are used both for regression and classification. Target value is predicted from the learned decision rules. Terminal nodes show the final / target value, whereas the branches represents the possible value set. In order to have an efficient DT, the dataset must be grouped into homogeneous subsets and DT should be constructed from discriminator attributes. Information Gain (IG) and entropy based methods are used to construct appropriate attribute set. ID3 and C4.5 (aka J48) algorithms have been introduced by J.R Quinlan which produce efficient DTs. C4.5 is an extension of ID3 algorithm [35].

Random Forests (RF), like DTs, are also used both for regression and classification. Random forests, first proposed by Ho [36] and later improved by Breiman [37], their work principle is based on ensembling, RFs constructs a huge forest of DTs by using a random subset of attributes/features. During model construction RFs produce multiple DTs and classification/regression is done by considering the results of each individual DT. RFs are efficient in handling high dimensional data which makes them a good candidate in NLP based applications. RFs give high accuracy also in bioinformatics and signal processing applications [38].

Naïve-Bayes is a probabilistic classifier based on Bayes' theorem, based on independence of features [39]. Mathematical expression is given in equation (2.2).

- c^* : predicted class
- h_{NB} : naive bayes function
- x : sample

$$c^* = h_{NB}(x) = arg\, max_{j=1,\ldots,m} P(c_j) \prod P(X_i = x_i \mid c_j) \tag{2.2}$$

Support Vector Machine (SVM) is primarily a classifier method that performs classification tasks by constructing hyper planes in a multidimensional space that separates cases of different class labels. SVM seeks to find a margin that separates all positive and negative examples.

Various kernels (linear, polynomial, RBF, sigmoid) are used in SVM. Parameter C controls the cost of misclassification on the training data. Small C makes the cost of misclassification low, large C makes the cost of misclassification high, thus forcing the algorithm to explain the input data stricter and potentially overfit. Large C gives low bias and high variance. Cross validation and resampling, along with grid search are good ways to find the best C. Gamma is a parameter for kernel function.

SVMs categorize the labeled samples by finding hyperplane that best divides the classes. SVM algorithm finds out the closest point (support vectors) to hyperplane while maximizing the distance between these points. This problem can be handled an optimization problem by aiming to minimize the equation 2.3. This minimization avoids over-fitting problems.

$$min(L(\beta) = \frac{1}{2} \parallel \beta \parallel^2), y_i(\beta^T x_i + \beta_0) \geq 1, \forall i \quad (2.3)$$

In equation 2.3, y_i represents labels for ith record, β represents weight vector, β_0 represents prejudice, x represents closest samples to hyperplane.

Generally samples are placed in complex order that is hard to separate. In order to separate samples usually hyperplanes are moved to higher dimensions using kernel functions. The kernel functions define a kind of distance or a similarity measure between the support vectors and the input instances. Although there is a huge list of kernel functions linear kernels, radial kernels and polynomial kernels are most commonly used kernel functions.

In linear kernels, the similarity (distance) is measured as dot product of instance and support vectors. Radial kernels enable to achieve complex regions on feature space. The last but not least kernel is polynomial kernel where kernel function provides curves.

2.3. Natural Language Processing

Three common sentiment analysis approaches are linguistic analysis, lexicon-based methods and full-text machine learning [16]. In many practical application, algorithms often used together [40, 41]. “Linguistic analysis” uses text grammatical form, context, idioms and negations to detect sentiment polarity [16, 42, 43]. A lexicon consists of words together with words polarity, and strength of those words [44]. In lexicon based approach, a dictionary that includes opinion words and their opinion rank annotated from negative to positive is needed. Sentistrength (SS) is a lexicon-based sentiment analysis tool/library developed by [16]. SS estimates the strength of positive and negative sentiment in short texts, reports two sentiment strengths:-1 (not negative) to -5 (extremely negative), 1 (not positive) to 5 (extremely positive). It can also report binary (positive/negative), trinary (positive/negative/neutral) and single scale (-4 to +4) results. SentiStrength was originally developed for English and optimised for general short social web texts but can be configured for other languages and contexts by changing its input files [45]. In addition to the main emotion lexicon, SentiStrentgh has negation words and booster words lexicon as well. Booster words increase or decrease the power of following opinion word's polarity rate.

2.4. Text Representation Methods

Text representation is vital for describing text messages to a computer algorithm. A key problem is how to effectively represent tweet messages posted by users to sentiment polarity classification pipeline. In our study, we have used "bag of words" and "word embeddings" to represent tweets.

The "bag of words" is commonly used in methods of document classification, where the frequency of separate words is used as an input feature for training a classifier [46]. The "bag of words" model is a simplifying natural language processing model in which a text is represented as an unordered collection of words. The sentence and/or document is represented with a vector of the words presence.

TF (Term Frequency) refers to the term Frequency of a word, the total count of the number of occurrences of a particular word in a document. Higher the value of TF, higher the weight for the feature. DF(Document Frequency) refers to number of documents in the collection that contain a specific word. Frequency of each word has been counted once for each document (different from TF) and ordered by frequency descending. IDF (Inverse Document Frequency) for a feature is calculated as follows : $IDF = log(N/DF)$ where, N refers to the total number of documents in the corpus. TF-IDF (Term Frequency - Inverse Document Frequency) TF-IDF score for a feature is computed as: $TD - IDF = TF * IDF$

In order to represent data to classifiers bag-of-words model have been used [46]. The sentence and/or document is represented with a vector of the words presence as shown in Table 2.1 and Table 2.2. Base on the problem type features can be word, word root or normalized word form. An enriched form can be using word combinations, in this approach a feature may consists of double words or more. Rather than using all word combinations, selecting most commonly used word pairs as feature will help to control feature vector dimension size.

Table 2.1: Feature Vectors based on Term Frequecy.

	Feature 1	Feature 2	Feature 3	....	Feature d
Tweet 1	1	0	0		1
Tweet 2					
..	0	5	8		0
Tweet N					

Table 2.2: Feature Vectors based on Presence/Absence Of Terms.

	Feature 1	Feature 2	Feature 3	....	Feature d
Tweet 1	1	0	0		1
Tweet 2					
...	0	1	0		0
Tweet N					

"Word embeddings" map phrases or words to vectors by considering the relationships between words [47]. The goal is to represent a word/phrase with a vector that contain real numbers. Word embeddings are successful in many NLP tasks [48]. Word Embedding principle is based on capturing context and semantic relationships between words. After a word or document is represented as a vector, it can be used in vector calculation, so vector operations like subtraction, addition or multiplication can be applicable. Documents, sentences or paragraphs contain many words, so word embedding vectors can be used to represent various type of texts. Each document can also be represented as vector by summing or multiplying words it contains. Blacoe studied multiple methods for document representation [49]. We have used modified versions of Blacoe models.

3. METHODS AND SYSTEM ARCHITECTURE

In this chapter, data collection and annotation, methods and common architectures used both in Turkish and English sentiment analysis are explained.

3.1. Data Collection and Annotation

Finding high quality annotated data is one of the challenges in NLP/ML problems, when it comes to Turkish it is even harder. Annotation is a time consuming task. In manual annotation process each document class is labeled manually. One common approach in such cases is to use crowd sourcing. The concept of crowd sourcing is based on observation that if a crowd of non-experts is asked an opinion, the aggregation of their individual opinions will be very close to the true value [50]. Another approach is to using limited training dataset samples as seed and continue with semi- supervised and unsupervised approach.

Banking sector data has been collected using the Twitter API and then annotated manually. We have collected tweets pertaining to leading banks and labeled to one of neutral, positive or negative polarity classes based on the message content.

Pre-collected datasets from Eksi Sozluk [51] and Kemik NLP Group [52] also have been used. Eksi Sozluk is a collaborative, information sharing web based dictionary and content platform [51]. The content is created by users in Turkish. Eksi Sozluk can be called as an information collaboration platform rather than classic dictionary. The information provided by users may be correct or not. Kemik is a natural language processing group in Computer Engineering Department at Yildiz Technical University [52]. Kemik has a rich set of datasets and open to all researchers.

Google News Trained word vectors are created from Google news corpus [53]. Word embeddings which are created from 100 billion words are 300- dimensional vectors. The model contains also misspellings of words, so majority of misspelled words vector representation can also be obtained from the pre-trained model. In our work, we have also used Google news Trained word vectors.

3.2. General System Architecture (GESA)

General solution consists two main categories: 1) Natural Language Processing (NLP) and 2) Machine Learning (ML) techniques. Figure 3.1 illustrates overall processing steps to estimate the sentiment polarity from tweet or documents. Tweets are captured from Twitter via an API supplied by Twitter. NLP tasks are applied in preprocessing. NLP tasks applied as a part of preprocessing and feature engineering and these tasks include Tokenization, Enrichment, Normalization, Morphological Analysis. Then, the feature extraction step determines specific features that will be in machine learning classification process.

Figure 3.1: General System Architecture.

Figure 3.2 illustrates preprocessing steps that have been used in Turkish based sentiment analysis pipeline. A tweet sentiment problem starts with a tweet collecting step. If the tweets are not labelled one must annotate (labeling) the tweets manually. In this study, as explained in dataset details tweet data has been both captured and also used from pre-collected tweet datasets.

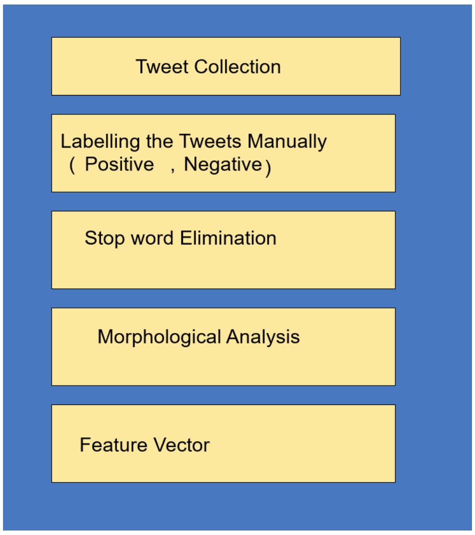

Figure 3.2: Preprocessing Steps.

3.3. Word Embedding Based Architecture (WEBA)

Word embeddings are derived by considering relationships between words in a corpus. Word embeddings convert each word to a vector [47]. Since each tweet, paragraph or document is colection of words then sentences, tweets or documents can also be represented with vectors. Ability to represent words with real number vectors enables us to apply vector calculus within our analysis. Blacoe studied multiple methods for document representation [49]. We have used modified versions of Blacoe models. We have considered to represent tweets as summation and/or multiplication of word vectors.

Word2vec, proposed by Mikolov *et al.*, is a group of related models that are used for word embedding, which is a technique for mapping words or phrases from the vocabulary to vectors of real numbers [49, 54]. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the

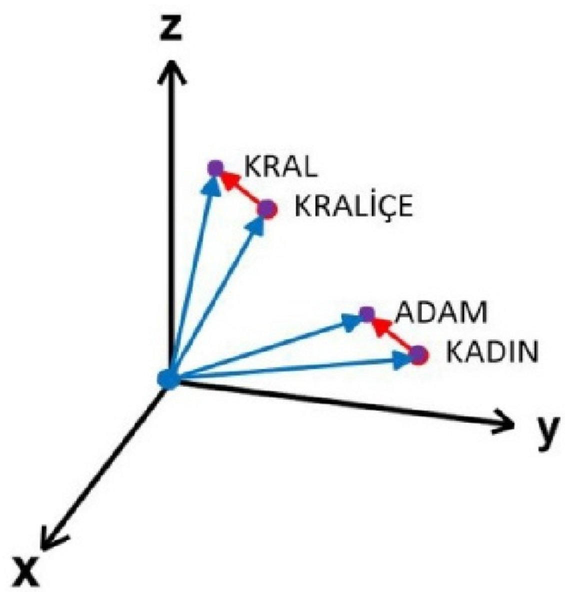

Figure 3.3: Word Vectors Representation

space. After a word or document is represented as a vector, then it can be used in vector calculation, so vector operations like subtraction, addition or multiplication can be applicable. Documents, sentences or paragraphs contain many words, so word embedding vectors can be used to represent various type of texts. A properly trained Word2vec model can calculate an operation like [king(kral)] - [man (adam)] + [woman (kadın)] and give the approximate result of [queen (kraliçe)] as shown in Figure 3.3. In this thesis, Word2vec model has been used to represent all words contained in corpus as vectorial entities.

Figure 3.4 illustrates the word vector generating pipeline steps that have been used in sentiment analysis word embedding pipeline.

In this work, a modified version of the sum representation method which is proposed by Blacoe is used [49] . The sum representation model, in its original state, is generated via summing the vectorial embeddings of words which a sentence contains. The related equations are given below with Equations 3.1, 3.2 and 3.3. For sake of easy interpretation symbols are explained below:

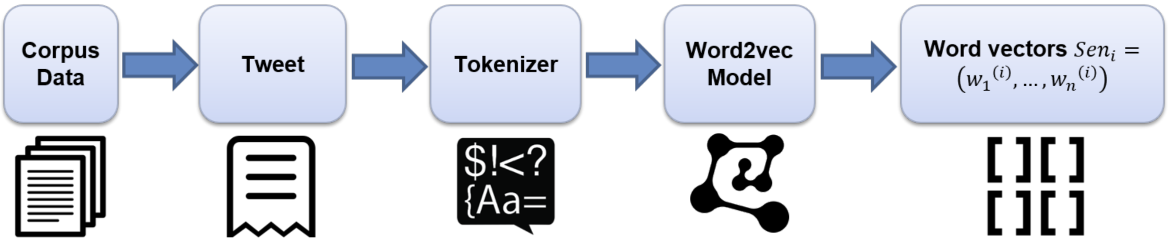

Figure 3.4: Generating Word Vectors.

n : Tweet word count (number of word in a tweet)

N: Total number of tweets

T_i: Tweet i

Twt: Tweet

$twtVec$: Tweet vector

w: word

$wdVec$: word vector

$$Twt_i = (w_1^{(i)}, \cdots, w_n^{(i)}) \tag{3.1}$$

$$twtVec[j] = \sum_{k=1,\ldots,n_i} wdVecwk[j] \tag{3.2}$$

$$twtVec[j] = \frac{\sum_{k=1,\ldots,n_i} wdVecw_k[j]}{n} \tag{3.3}$$

Another method is based on multiplication, the formula is given by Equation 3.4. Multiplication method, generates sentence vectors via multiplication of vectorial representation of each word included in the sentence. Mathematical expression is given in Equation 3.4.

$$twtVec[j] = \prod_{k=1,\ldots,n_i} wdVecwk[j] \tag{3.4}$$

Figure 3.5 illustrates usage of word vectors and classifiers together. Word vectors are fed to classifiers to detect sentiment polarity.

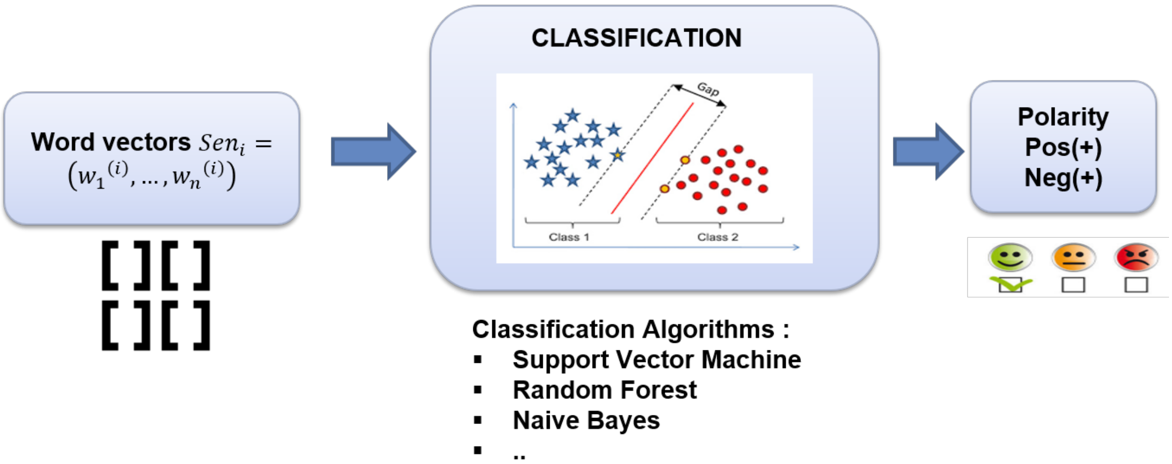

Figure 3.5: Word Embedding and Classifiers.

Word2vec models can be generated from corpus under consideraton or a previously prepared word2vec model (like Google news trained word2vec model) can be used. In this thesis both options have been used.

3.4. Long Short Term Memory Based Architecture (LOBA)

Recurrent Neural Networks have long-term dependency problems. LSTM networks avoids these long-term dependency problems. They are efficient in capturing long-term features and dependencies [34]. LSTMs use state information and for every input LSTMs decide which data is going to be throw away and what data is going to be stored. LSTMs have cells that are working as processing units to process state and inputs. Cells (aka memory units) process previous states, current input and produce an output.

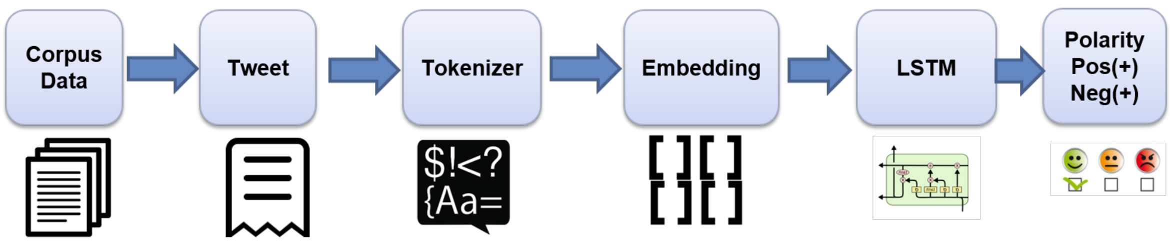

Figure 3.6: Long Short Term Memory Pipeline

Figure 3.6 illustrates Long Short Term Memory with embedding preprocess pipeline steps that have been used in in sentiment analysis. Tweet dataset has been preprocessed before feeding into the LSTM classifier. Punctuations have been removed, all content have been converted to lowercase.

3.5. Used Tools and Software Packages

During application implementation and tests we have used many software packages and toold including Deeplearning4j, Keras, Zemberek.

"Deeplearning4j" [55] is a deep-learning library written for Java and Scala. In this thesis, we have used Deeplearning4j for semeval task. "Keras" is developed by Chollet [56] in python. Keras uses tensorflow and Theano libraries in backend. We have used keras for LSTM related tasks. Majority of Turkish based NLP tasks use “Zemberek” since it is open source and platform independent [57]. We have used Zemberek as part of preprocessing steps like obtaining word roots.

4. SENTIMENT ANALYSIS IN ENGLISH

Sentiment analysis in English has been studied using two independent datasets. The first one is taken from Semeval 2017 and the second one is collected for political opinion/sentiment prediction studies.

This chapter starts with describing the approaches implemented for SemEval. We start by explaining our solution for SemEval-2017 Task four: SA in Twitter. We have participated in Semantic Evaluation Subtask A: Message Polarity Classification subtask and implemented two different systems. The first system is based on word embeddings, uses word embeddings for feature representation and vectorization, and SVM, RF and NB algorithms for the classification of tweets into one of positive, negative and neutral classes. The second system is uses Long Short Term Memory (LSTM) which is a specialized sequential Recurrent Neural Networks classifier and uses word indexes as sequence of inputs. The chapter continues with tweet based political orientation prediction. We have used various machine learning algorithms that can predict and automatically classify whether a tweet belongs to republican or a democrat. We have used a Twitter dataset that was produced by democrat and republican voters. This work covers political opinion/sentiment tendency estimation based on Twitter messages of voters.

4.1. SemEval Problem Definition

This section briefly introduces the problem definition and formulation for SemEval which is an ongoing series of evaluations of computational semantic analysis systems, organized under management of SIGLEX, the Special Interest Group on the Lexicon of the Association for Computational Linguistics (ACL). SemEval 2017 has many tasks for sentiment analysis for Twitter [21]:

(i) Message Polarity Classification: Given a message, classify whether the message is of positive, negative, or neutral sentiment.

(ii) Topic-Based Message Polarity Classification: Given a message and a topic, classify the message on B) two-point scale: positive or negative sentiment towards that topic C) five-point scale: sentiment conveyed by that tweet towards the topic on a five-point scale.

(iii) Tweet Quantification: Given a set of tweets about a given topic, estimate the distribution of the tweets across D) two-point scale: the "Positive" and "Negative" classes E) five-point scale: the five classes of a five-point scale.

4.2. System Decsription

We have developed two systems for Semeval, the first one is "Word Embedding based System" and the second system is "LSTM Based System".

Word embedding sytem flow is given in Figure 4.1. As shown in Figure 4.1, the system uses both word2vec model vectors and google news based pre-trained vectors. Word2vec uses all SemEval datasets to create word embeddings. We have used and tested both forms to see effect of the corpus.

The pipeline of the second system (LSTM Based System) consists of many steps: reading Tweets from Semeval datasets, preprocessing Tweets, representing each word with an index, then representing each Tweet with a set of word index sequence and training a LSTM classifier with sequence index array. The Flowchart of this system is shown in Figure 4.2.

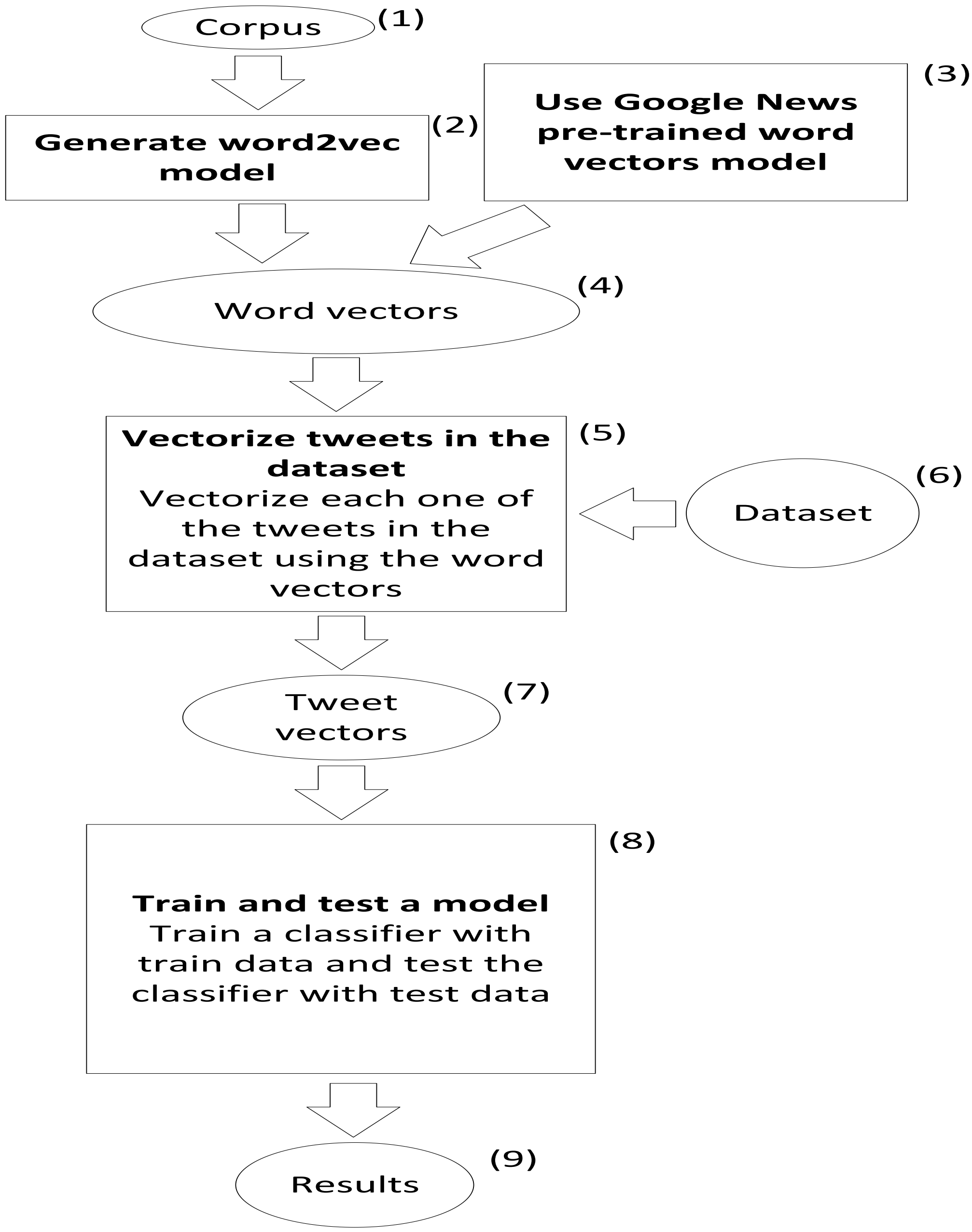

Figure 4.1: General flow of the word embedding based system.

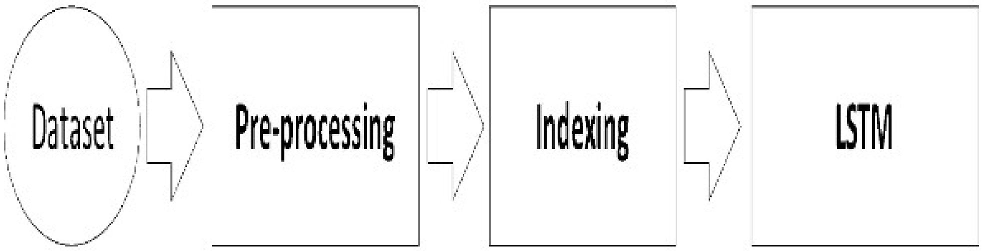

Figure 4.2: LSTM based system pipeline.

4.3. Methods and Tools

In this section, we explain methods and tool we have used. We have applied preprocessing steps, used indexing and word embedding as well as classification algorithms. Basic pipelice is shown in Figure 4.2. The steps are sequential and should be applied as shown order.

In pre-processing step, basic operations have been applied to the dataset before feeding it into the LSTM classifier. Deeplearning4j library [55] has been used to remove punctuations from tweets, and convert all content into lowercase.

In word embedding process, Word2vec models have been generated both from Google news and from Semeval datasets. Word2vec models have been generated with the parameters shown in Table 4.1.

Table 4.1: Parameters for word2vec generation stage.

minWordFreq	MIN_WORD_FREQ = 5 (Minimum word frequency)
iterations	NETWORK_ITERATION = 25/50 (Number of batch iterations).
layerSize	FEATURE_VECTOR_DIMENSION_SIZE = 300/600/900 (Word embedding vector dimensions size).

In indexing step, each word is assigned an index, and each tweet is defined as a set of index numbers. This representation allows each tweet to be represented in a sequential manner. LSTM networks are fed through this indexed word set, so that LSTM can consider word sequence and order.

In classification step SVM, RF, NB and LSTM have been used as classifiers. SVM has been configured with the following parameters:

$$Kernel = PolyKernel, batchSize = 100$$

Random Forests classifier has been used with the following parameters:

$$bagSizePercent = 100, batchSize = 100$$

LSTMs are different from SVM, NB and RF in the sense that they process input data sequentially. Before feeding the "Long Short Term Memory Recurrent Neural Net", input data must be prepared to appropriate format as shown in Figure 4.2. In this thesis, categorical cross entropy and softmax function have been used with LSTM. Developed model parameters are given in Table 4.2 and LSTM Classifier layered structure is shown in Figure 4.3.

Table 4.2: Parameters for classifier stage.

max_features	86000 : Maximum integer value of indexed dataset.
maxlen	25 : Indexed tweets padded into this value.
batch_size	32
model	Sequential(): sequential model
Embedding	max_features : size of the vocabulary
Embedding	max_features : Input dimension, size of the vocabulary.
LSTM	128 : dimension of the internal projections and the final output dropout
Dense	3 : Output dimensions.
Activation	'softmax' : Normalized exponential function
loss	'sparse_categorical_crossentropy'
optimizer	'adam' : Adam optimizer.

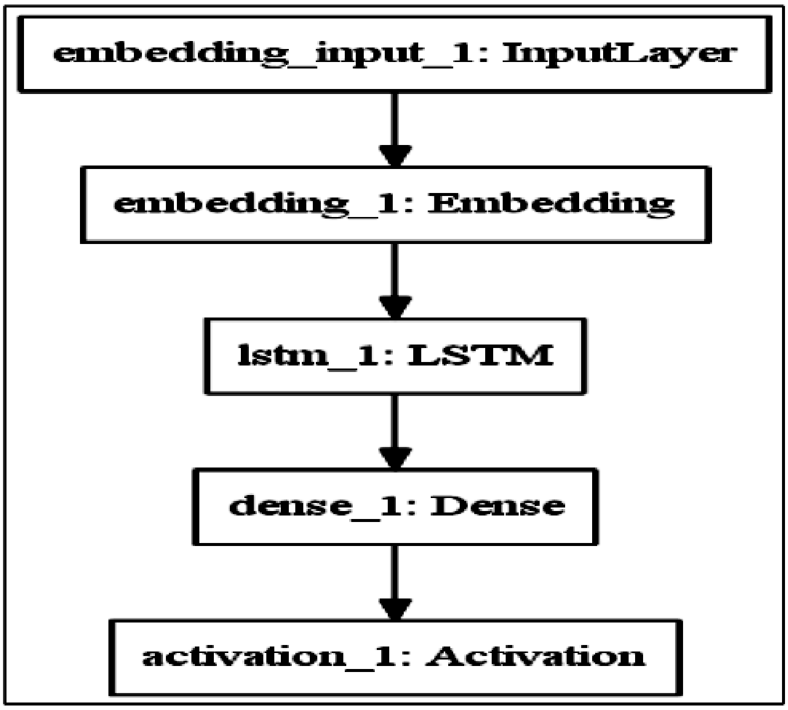

Figure 4.3: LSTM classifier layered structure.

4.4. Experimental Study in Semeval

4.4.1. Dataset

SemEval tasks have datasets with Twitter messages on a range of topics including products, events, mixture of entitites [21]. Datasets belonging to previous years are available for training, development and testing the systems.

4.4.2. Test Setup

We have tested various setups to achieve the highest accuracy rate. 5 test cases have been tested with different parameters as shown in Table 4.3. In the first three test cases (Test 01,02, and 03) word2vec model has been trained with SemEval corpus. Tests have been done with SVM, RF and NB. Test 05 is conducted with LSTM classifier. In Test 05, we have fed LSTM with maximum 30 words, since we have assumed that a tweet does not contain more than 30 words. For each word we have assigned an integer index number. During creation of index numbers we have scanned all dataset and we have represented each word with a unique number. LSTM input vector size is 30. When a tweet contains less than 30 words, we have used padding.

Table 4.3: Test cases and setup configuration

Test No	**Word Vectors**	**Dimension Size**	**SVM**	**RF**	**NB**	**LSTM**
1	SemEval	600	√	√	√	
2	SemEval	300	√	√	√	
3	SemEval	900	√	√	√	
4	Google News trained	300	√	√	√	
5	Word index	30				√

4.4.3. Tests with Word Embedding, SVM, RF, NB and LSTM

Purpose of the first three tests was observing impact of the parameter feature vector dimension size's and classifier type impact on the general performance. We have used SemEval training and test datasets pertaining to 2013, 2014, 2015 and 2016 years to construct SemEval word2vec model. The tests have been done using this SemEval cumulative dataset.

In order to evaluate also classifiers we have repeated our tests with SVM, RF and NB. We have reported results for each classifier. Google news pre-trained word vectors gave better results compared to others and has positive impact on accuracy rates for all classifiers. Results obtained from the classification tests are shown in Table 4.4.Results are reported after 10 fold cross validation. SVM and Google news trained vectors combination gave better result compared other classifier and corpus vectors. We have obtained 62.8% accuracy rate.

Table 4.4: Results obtained from tests.

Test No.	Word Vectors	Dim.Size	SVM Acc. %	RF Acc. %	NB Acc. %
1	SemEval	600	58.3	54.4	52.3
2	SemEval	300	57.3	45.7	51.8
3	SemEval	900	58.7	53.7	51.7
4	Google News trained word vectors	300	**62.8**	57.2	53.1

Keras library is used to train and test LSTM Recurent Neural Net. Test 05 id done with LSTM classifier on SemEval cumulative dataset and 62.6% accuracy rate has been achieved. Using SemEval 2017 test data we have achieved the following scores: Average = 0.587, Average R = 0.605 and Accuracy = 0.603. In this last test, test dataset has been shared with us as unlabeled by SemEval committee. By running our code and algorithms we have labeled the test dataset and share the labeled dataset with SemEval committee, then the results are shared with us by organization committee. We have been invited to describe our work and algorithms after the evaluation process. Our paper has been published in proceedings of 11th international workshop on semantic evaluation [8].

4.4.4. Discussion

We have obtained the best accuracy rate (62.8%) with SVM classifier and Google News pre-trained word embedding vectors. Accuracy rate has been achieved when our model (Word Embedding + SVM) is applied to 2016 and previous years SemEval datasets. On the Semeval 2017 Test Dataset by using the same model (the first system structure), we have achieved 60.3% accuracy rate with the following scores scores : Average = 0.587, Average R = 0.605 and Accuracy = 0.603.

In order to create a better system different approaches can be applied. One alternative to implement more enhanced system could be to deploy word embedding vectors to classifiers like Convolutional Neural Networks (CNN), Recurrent Tensor Neural Networks, and combining LSTM and CNNs etc. As an alternative, another method can be considering feature level fusion of hand crafted features and lexicons (bag of words, n-grams) with word embedding based vector representation.

4.5. Experimental Study in Political Opinion/Sentiment Analysis

Classification of users reviews about a concept or political view may bring different opportunities. This section addresses the task of political orientation classification based on Twitter data. A twitter dataset that was produced by republicans and democrat voters and various machine learning algorithms that can predict and automatically classify whether a tweet belongs to republican or a democrat have been studied.

4.5.1. Dataset

We have used a Twitter dataset that was produced by republicans and democrats (courtesy of Eser Aygun, ITU) on Twitter platform. Each line is a tweet, the first word shows the label. In the dataset there is noise. By noise we mean: some words are misspelled, poor spelling, poor punctuation, poor grammar, some words may be omitted, word or sentence ambiguity, abbreviations.In the dataset all properties of Twitter language has been used. Users often include Twitter user names in their tweets in order to direct their messages. Users usually include URL links in their tweets. When a tweet contains an URL, content or category of that URL page may increase sentiment analysis performance.

To do basic analysis of the dataset we use a Tokenizer software, StanfordCoreNLP [58]. This software takes raw English language text as input and gives the base form of words. Dataset properties summary is given in Table 4.5. Table 4.6 shows most frequently used words for republicans and Table 4.7 shows for most frequently used words for democrats.

Table 4.5: Dataset Properties Summary.

Total # of tweets	165352
# of Republicans	74523
# of Democrats	89465
# of non-opinion	1364
# of tweets containing URL	96904
# of class	3
#of distinct word/entity	109988

4.5.2. Tests with BoW and SVM

Tweets have been represented with BoW. Information Gain (IG) and mRMR feature selection algorithms have been used. Results are shown in Table 4.8.

4.5.3. Tests with Word Embedding and SVM

Woodwork models have been used in two different models. In the first approach word2vec model has been generated from political opinion/sentiment dataset. In the second approach googlenews pre-trained word2vec model has been used. Results are shown in Table 4.9.

Table 4.6: Frequently Used Words for Republicans.

WORD	**Republican**	**Democrat**	**None**
Obama	11819	6200	121
#tcot	8500	468	39
Romney	4955	8354	118
#teaparty	3795	68	1
#twisters	3703	0	0
2012	3685	6306	89
New	2658	3882	124

Table 4.7: Frequently Used Words For Democrats

WORD	Republican	Democrat	None
Romney	4955	8354	118
2012	3685	6306	89
Obama	11819	6200	121
#p	2371	5231	72
New	2658	3882	124
#us	6	3724	2
Ryan	2318	3655	90
president	1734	3262	73
Mitt	1222	3193	32

Table 4.8: Political Opinion / Sentiment Test Results for BoW and SVM

Feature Selection Method	**Accuracy %**
IG	63.32
mRMR	62.98

Table 4.9: Word Embedding and SVM Results for Political Opinion / Sentiment

Corpus	**Accuracy %**
Political Tweet Dataset Corpus based Word2vec Model	62.81
Googlenews Word2Vec Model	61.65

4.5.4. Test with LSTM

LSTM classifier is feeded with word indexes rather than BoW or word embedding model vectors. Result is depicted in Table 4.10.

Table 4.10: Classification with word indexes and LSTM

Tweet Representation Method	Accuracy %
Word Index	77.92

4.5.5. Discussion

Table 4.11: General Results for Twitter Political Opinion Sentiment Prediction

Classifier	**Tweet Representation Method**	**Accuracy %**
LSTM	Word Indexes	77.92
SVM	BoW	63.32
SVM	Word Embedding	62.81

Word index based LSTM performed better compared to BoW and Word2Vec generated model vectors as depicted in Table 4.11. 77.92% accuracy rate has been achieved with Word index based approach.

5. SENTIMENT ANALYSIS IN TURKISH

This chapter introduces Sector Based Sentiment Analysis Framework which is proposed to predict sentiment polarity in different sectors.

5.1. Problem Definition and Formulation

This section briefly introduces the problem definition and formulation. There are many studies on sentiment analysis focused on English language whereas there are only limited number of sentiment analysis studies for Turkish language. The rise of social media usage enabled easy and versatile content creation. Foursquare, Instagram, Whatsapp, Facebook, Tango, Messenger, Skype like platforms are commonly used to communicate and share opinions. Tremendous increase in usage of social media content has created new opportunities to study public sentiment, with Twitter being especially popular for research due to its ease of access to its content and high volume.

In Twitter usage, generally some words are misspelled, there is poor spelling, poor punctuation, poor grammar, high abbreviation usage. Also some words may be omitted and word or sentence ambiguity is encountered frequently. The same word can be written with different characters. Users often include Twitter user names and URL links in their tweets. When a tweet contains an URL, content or category of that URL page may increase sentiment analysis performance. Hashtag is also used (a tag prefixed with a # character to identify topics). Twitter messages have also emoticons.

Turkish social media text mining has many challenges. Turkish is an agglutinative language. Due to its agglutinativeness many forms of same word can be encountered in texts. Mobile phone keyboards are usually English based and special Turkish specific characters like 'ğ', 'ş', 'ü', 'ö' are not used and replaced with close English characters whereas some keyboards have Turkish character support. Turkish is also a language where sarcasm and irony is commonly used.

Data collection is one of the biggest challenges in this task. Finding rich annotated data in Turkish is a hard task. Existing limited training data is highly noisy.

5.2. General Solution Pipeline (GSP)

5.2.1. GSP Architecture

General solution consists two main categories: 1) Natural Language Processing (NLP) and 2) Machine Learning (ML) techniques. NLP tasks applied as a part of feature engineering and these tasks include Tokenization, Enrichment, Normalization, Morphological Analysis steps as shown in Figure 5.1.

Figure 5.2 depicts the proposed solution framework. Data has been retrieved from different sources. Based on the source data, the sector is identified. Once the sector is identified, NLP based preprocessing pipeline is applied.

5.2.2. Preprocessing

Preprocessing steps including tweet collection, labelling the tweets manually, deascifying the tweets, morphological analysis and feature vector creation in arff format Figure 5.2 In preprocess phase punctuation handling has been done (Table 5.1), words has been tokenized, 1-gram and 2-gram based features has been obtained.

Majority of Turkish based NLP tasks use Zemberek since it is open source and platform independent. We have also used Zemberek Library to obtain word roots.

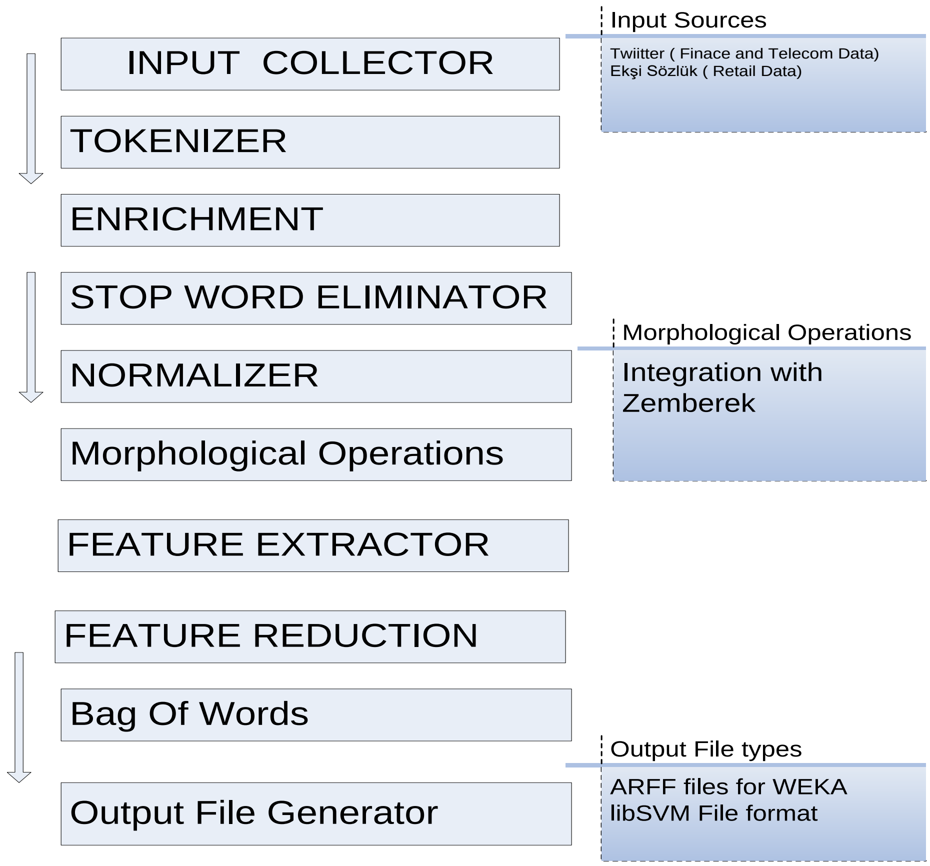

Figure 5.1: Preprocessing Steps.

Figure 5.2: Solution Components.

Table 5.1: Punctuation, numbers and URL Handling.

Punctuation	**Feature**
$:-)$	SMILEY_POSITIVE
$:-))$	SMILEY_POSITIVE
$:-($	SMILEY_NEGATIVE
$:-(($	SMILEY_NEGATIVE
$(,),<,>,?,!,.$	Omitted
$[0-9]$	Omitted
$http://xxx.xxx$	HTTP_URL
@*name*	NAMED_ENTITY_name
#hashtag	HASHTAG_hashtag

We have used pos tags to enrich feature set and to differentiate polarity of each word. Table 5.2 depicts some examples, for example a word belgelemek in Turkish (to document) is used as a input feature as belgele+ verb + positive whereas belgelememek (not to document) is used as belgele+verb+negative, tatlı (sweet) is used as tatlı+adjective and haksızlık (unjustness) is used as haksızlık + noun + negative after morphological analysis.

Table 5.2: Morphologically enhanced features.

memnunum	*memnun + verb + positive*
belgelemek	*belgele + verb + positive*
belgelememek	*belgele + verb + negative*
tatlı	*tatlı + adjective*
kocaman	*kocaman + adjective*
haksızlık	*haksizlik + noun + negative*

5.2.3. Feature Representation

In order to represent data to classifiers, bag-of-words model have been used [46]. The sentence and/or document is represented with a vector of the words presence as shown in Chapter 2 Table 2.1 and Table 2.2.

5.2.4. Feature Reduction and Selection

We have used Stanford Library to obtain base words. Clean tweet data file has been parsed and each tweet has been represented as a list of root words at first step. We are not dealing with tweets directly in classification, but firstly we applied pre-processing step which removes all the stop-words, and then prepared input vectors.

In order to obtain the appropriate feature set following methods has been used collaboratively: TF, DF, IG, mRMR. The goal is to select a the right feature subset that characterizes the target classification variable. We have calculated TF, DF scores for all words in training data. Then we have eliminated stop words including and, or, an, as, by, he, his, me, who etc. We have used a free stop-word list [59]. We have prepared weka input datasets (arff file) based on the term frequency, document frequency. Based on TF and DF scores we have decided to consider words with the highest scores as feature vector. In TF, frequency of each word has been counted and ordered by term frequency and descending. Based on the DF, TF score at first step we have selected 300 features with the highest score. The highest 300 attributes score is significantly larger compared to the rest of attributes. In order to focus on the right feature set we have used Information Gain and mRMR methods. Both IG and mRMR have been explained in Chapter 2. After applying both IG and mRMR feture set, dimension decreased to approx. 170 (TF+IG 169, DF+IG 170) for IG and 220 for (TF+mRMR 227, DF+mRMR 192).

5.3. Word Embedding based System Pipeline (WESP)

This section explains word embedding based system pipeline for Turkish language. Pipeline has been applied to finance, telecom, football and retail sectors. By using SVM, RF, NB and word embedding models tweets pertaining to various sectors has been vectorized, classified and results have been compared.

In this study, two word representation have been considered, sum based and multiplication based representation. Sum Method and Multiplication based methods are explained in chaper 2.

5.4. Experimental Study

Several tests are conducted on different datasets. The details of these datasets are given in section 5.4.1 and test results are given in section 5.4.2.

5.4.1. Data Collection and Datasets

There exists a rich set of sector domains including auto, energy, banking, telecom, healthcare, technology, retail and insurance. In this work we have concentrated only on Banking, Telecom and Retail domains. We have used both Twitter data and Eksi sozluk[51] to train classifiers. We have gathered data from different sectors including finance, retail and telecom: 1) Finance Sector Dataset(collected from twitter), 2) Retail Sector Dataset (collected from eksi sozluk), 3) Telecom Sector Dataset (collected from Kemik [52]), 4) Aggregated Dataset

“Tweet Collection and Annotation”: In order to prepare finance sector dataset Tweet messages have been retrieved via API from Twitter. Twitter has an open service that is called as Application Programming Interface (API). Tweets can be queried with query items through API services. We have developed our own tweet retrieval module. The tweet retrieval module has been developed by Java. After retrieving tweets we have annotated them manually for banking and football sector.

Eksi Sözlük is a collaborative ’dictionary’ based on the concept of user contribution [51]. Eksi Sözlük is not a official dictionary in the strict sense; users may use slang, abbreviations etc. Eksi Sozluk is used for knowledge and sharing on various topics ranging from everyday life concerns to deep science topics. Retail data has been gathered from Eksi Sozluk. ‘Retail sector ‘Products Dataset” contains user feedback tweets about multiple technological products and their manufacturers. “Kemik Dataset” is derived from “3000 Tweets” dataset from Kemik Natural Language Processing Group [52], which has user feedback tweets about a mobile GSM operator company. “Football Dataset” contains user feedback tweets about Turkish Football. “Bank Dataset” contains customer feedback tweets about banks and their services.

Table 5.3: Turkish Datasets

Sector	**Total Records**	**Positive Instance**	**Negative Instance**
Retail (*EksiSozluk*)	1707	865	842
Telecom (*Kemik/Twitter*)	2043	1287	756
Football(*Twitter*)	1340	833	507
Banking (*Twitter*)	718	604	114
TOTAL	5808	3589	2219

Table 5.4: Word2vec Model Training Datasets.

	Total number of Tweets
Retail + Telecom + Football + Banking	5808
Kemik B**	158.885
TOTAL	164.693

5.4.2. GSP Tests and Results

SVM, NB, MNB, Decision Trees and kNN algorithms have been used for classification during tests. For validation 10-fold cross validation technique has been used. Test cases are shown below:

(i) Tests with Support Vector Machine
(ii) Tests with Multinomial Naive Bayes
(iii) Tests with kNN like Similarity Based Class Assignment Approach

As a robust classifier "SVM" conducts classification process by constructing hyper planes in a multidimensional space that separates instances of different class members. libSVM package has been used within java codes to conduct SVM tests. Dataset has been transformed to the svm package format. In order to find right SVM configuration parameter setup, various kernels (linear, polynomial, RBF, sigmoid) and various penatly factor C and gamma parameters have been tested. C controls the cost of misclassification on the data. Small C values make the cost of misclassification low, while large C values make the cost of misclassification high. Large C values give low bias and high variance. Cross validation and resampling, along with grid search are good ways to find the best C. Gamma is a parameter for kernel function. The results are depicted in Table 5.5.

Table 5.5: SVM Results by with various Kernel, C and Gamma parameters

Dataset	**SVM Kernel**	**C**	**Gamma**	**Accuracy**
Finance	Linear	0.1	8	91.19
Finance	Polynomial	0.1	8	91.19
Finance	RBF	0.1	8	91.19
Finance	Sigmoid	0.1	8	91.19
Retail	Linear	100	0.25	64.04
Retail	Polynomial	1	8	64.04
Retail	RBF	10	8	64.04
Retail	Sigmoid	0.1	8	64.04
Telecom	Linear	10	0.125	77.85
Telecom	Polynomial	10	0.125	77.85
Telecom	RBF	10	0.125	77.85
Telecom	Sigmoid	10	0.125	77.85
Cumulative	Linear	1	0.125	82.04
Cumulative	Polynomial	1	0.125	82.04
Cumulative	RBF	1	0.125	82.04
Cumulative	Sigmoid	1	0.125	82.04

"Multinomial Naive Bayes (MNB)" is a kind of NB classifier based on applying Bayes theorem. MNB classifier assumes the input features are independent. NB considers the probability of each word appearing in a document for classification. This probability valus is calculated based on training data [60]. The occurrence of words is believed to be independent of each other. MNB is a specialized version of Naive Bayes that is designed especially for text items.

In "kNN like Similarity Based Class Assignment Approach" annotated training set is used as a reference. Candidate Tweet Class is assigned based on distance measure both to positive class tweets and negative class tweets as shown in Figure 5.3. Average positive similarity is calculated and tweet is assigned to class with more average similarity. Results are depicted in Table 5.6.

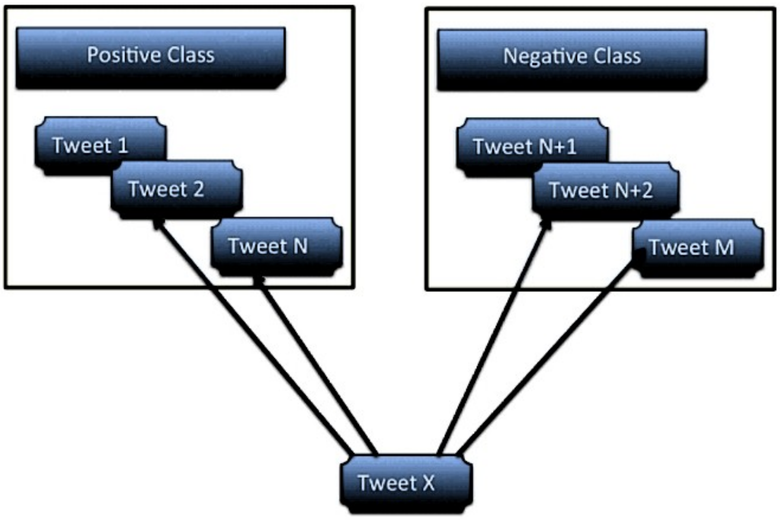

Figure 5.3: Tweet class assignment.

Table 5.6: Comparison of Methods Results.

Classifier	**Accuracy %**
SVM	82.04
Multinomial Naïve Bayes	76.12
Similarity Based Method	65.06

SVM worked better than Naïve Bayes and similarity based class assignment methods. For finance dataset 91.19% and 82.04% for aggregated dataset accuracy rate have been achieved as shown in Table 5.6. The results suggest that sector based approach is better than general sentiment analysis approach, identifying sector segment and then applying machine learning pipeline increases accuracy rate.

5.4.3. WESP Results

Results are given above in Table 5.7. Tests have been done with 10 fold cross validation using Support Vector Machine and Random Forests classification algorithms. Retail (Products), Banking, Football and Telecom datasets all have been tested individually. In addition to seperate individual datasets, combination of all also have been tested. Corpus effect also has been tested by using different corpora. Corpus based results are shown in Table 5.8.

Table 5.7: Word Embedding based Results with RF and SVM Classifiers in Turkish.

Test #	Dataset	Random Forest (%)	Random Forest Accuracy (%)	Support Vector Machine Accuracy (%)	Support Vector Machine Accuracy (%)
	Data	Sum Model	Multiplication Model	Sum Model	Multiplication Model
1	Retail	59.63	52.02	63.68	51.03
2	Banking	86.49	84.82	89.97	84.12
3	Football	84.93	81.19	84.02	72.46
4	Telecom	73.03	65.59	73.86	63.14
5	Total	62.93	67.13	74.6	63.31

Table 5.8: Corpus Comparison Results.

Test #	Data	Standard Corpus (Product + Kemik A + Footballl+ Banking) Accuracy (%)	Enriched Corpus (Products + Kemik A + Kemik B + Football + Banking)
1	Retail	62.57	63.68
2	Banking	89.41	89.97
3	Football	82.91	84.93
4	Telecom	72.49	73.86
5	Total	73.19	74.60

Results confirms that sector based SA is more successful compared to general, non-sector specific tweets. In terms of accuracy percentage, the Banking sector accuracy rate is 89.97%, Football is 84.02%, Telecom is 73.86%, Retail is 63.68% and for overall 74.60% have been obtained.

5.4.4. LSTM Based Results

Tests have been done with 10 fold cross validation using LSTM. Retail (Products), Banking, Football and Telecom datasets all have been tested individually. In addition to seperate individual datasets, combination of all also have been tested. LSTM Classifier is feeded with word indexes. Results are given in Table 5.9.

Table 5.9: Turkish Datasets

Sector	**Total Records**	**LSTM Acc. %**
Retail (*EksiSozluk*)	1707	64.40
Telecom (*Kemik/Twitter*)	2043	72.75
Football(*Twitter*)	1340	89.58
Banking (*Twitter*)	718	97.58
TOTAL	5808	76.47

5.5. Discussion

Although it is easy task to access social media data, labeling data manually and/or finding an annotated dataset to use in supervised training phase is a hard task. We have collected and labeled our own datasets as well as using pre-prepared datasets in our tests. We have conducted many tests on Turkish datasets to compare methods. The results suggests that sector based approach is better than general sentiment analysis approach, identifying sector segment initially and then applying machine learning pipeline increases accuracy rate. Experiments also showed that applying word embedding and lstm based sequential approaches have comparable and /or better results compared to handcrafted based features. Word embedding based vector representations are good in detecting language features and word relations for Turkish language. Word embedding based methods helped us to develop sentiment analysis pipeline without using a lexicon which suggest us in a multilingual need, the same system architecture can be used. We have created our own word embedding vectors but we have also used pre-trained vectors. We have noticed that using rich pre-trained word vectors also have positive effect in terms of increasing accuracy rate and decreasing computational time. Using word embeddings which is a type of unsupervised learning from a large unlabeled articles, news or comments is better than using or preparing labeled data which will take also huge effort and time.

6. CONCLUSION

In the context of this thesis, we have studied the effectiveness of applying ML and statistical NLP techniques collaboratively to twitter messages based sentiment classification problem both for Turkish and English. We have decided to work also on Turkish, because although there are a lot of studies for English language, Turkish language based sentiment analysis studies are limited and this problem has not been solved for Turkish language at the moment. The thesis covers the work on Turkish tweet sentiment analysis, sector based sentiment analysis, English tweet sentiment analysis, as well as political opinion prediction. We have aimed to answer sentiment analysis related questions within this thesis work. Main drivers in this thesis are investigating the effect of collaboration of ML and NLP techniques, sector based sentiment analysis, text representations and sequential approaches versus old school ML techniques.

Although it is easy task to access social media data labeling data manually and/or finding an annotated dataset to use in supervised training phase is a hard task. We have collected and labeled our own datasets as well as using pre-prepared datasets in our tests.

We have conducted many tests on Turkish and English datasets to compare efficiency of methods. The results suggests that sector based approach is better than general sentiment analysis approach for some sectors, identifying sector segment initially and then applying machine learning pipeline increases accuracy rate. The results also suggest that machine learning based approaches are good enough to compete with linguistic and lexicon based approaches so that the proposed architecture and pipeline can be adapted for different languages easily.

Experiments also showed that applying word embedding and LSTM based sequential approaches have comparable and /or better results compared to handcrafted based features. Word embedding based vector representations are good in detecting language features and word relations for Turkish language. Word embedding based methods helped us to develop sentiment analysis pipeline without using a lexicon which suggest us in a multilingual need, the same system architecture can be used. We have created our own word embedding vectors but we have also used pre-trained vectors. We have noticed that using rich pre-trained word vectors also have positive effect in terms of increasing accuracy rate and decreasing computational time. Using word embeddings which is a type of unsupervised learning from a large unlabeled articles, news or comments is better than using or preparing labeled data which will take also huge effort and time.

It is evident that in order to increase the overall performance there is a need for future work. We have focused just on machine learning algorithms. As a future work, this study also may be improved by lexicon approach and linguistic analysis methods. In order to create a better system different approaches can be applied. One alternative to implement more enhanced system could be to deploy word embedding vectors to classifiers like Convolutional Neural Networks (CNN), Recurrent Tensor Neural Networks (RTNN), and combining LSTM and CNNs etc. As an alternative, another method can be considering feature level fusion of hand crafted features and lexicons (bag of words, n-grams) with word embedding based vector representation. Since finding a rich annotated proper dataset is hard and manually annotating process is costly, unsupervised and graph algorithms also should be considered.

REFERENCES

1. Bethard, S., H. Yu, A. Thornton, V. Hatzivassiloglou and D. Jurafsky, "Automatic extraction of opinion propositions and their holders", *2004 AAAI spring symposium on exploring attitude and affect in text*, Vol. 2224, 2004.

2. Thomas, M., B. Pang and L. Lee, "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts", *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 327–335, Association for Computational Linguistics, 2006.

3. Bo, P., L. Lillian and V. Shivakumar, "Thumbs Up?: Sentiment Classification Using Machine Learning Techniques", *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pp. 79–86, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, `https://doi.org/10.3115/1118693.1118704`.

4. Pang, B. and L. Lee, "Opinion mining and sentiment analysis", *Foundations and Trends® in Information Retrieval*, Vol. 2, No. 1–2, pp. 1–135, 2008.

5. Turney, P. D. and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association", *ACM Transactions on Information Systems (TOIS)*, Vol. 21, No. 4, pp. 315–346, 2003.

6. Bikel, D. and I. Zitouni, *Multilingual natural language processing applications: from theory to practice*, IBM Press, 2012.

7. Ayata, D., M. Saraçlar and A. Özgür, "Turkish tweet sentiment analysis with word embedding and machine learning", *25th IEEE Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2017.

8. Ayata, D., M. Saraclar and A. Ozgur, "BUSEM at SemEval-2017 Task 4A Sentiment Analysis with Word Embedding and Long Short Term Memory RNN Ap-

proaches", *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 777–783, 2017.

9. Ayata, D., M. Saraçlar and A. Özgür, "Political opinion/sentiment prediction via long short term memory recurrent neural networks on Twitter", *25th IEEE Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2017.

10. Ayata, D., M. Saraclar and A. Ozgur, "Sector based Sentiment Analysis Framework for Social Media via Machine Learning and Natural Language Processing Approaches", *Proceedings of the Fourth International Symposium on Engineering, Artificial Intelligence and Applications (ISEAIA 2016)*, 2016.

11. Rao, D., D. Yarowsky, A. Shreevats and M. Gupta, "Classifying latent user attributes in twitter", *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pp. 37–44, ACM, 2010.

12. Mihalcea, R. and D. Radev, *Graph-based natural language processing and information retrieval*, Cambridge University Press, 2011.

13. Amasyalı, M. F., S. Balcı, E. Mete and E. N. Varlı, "Türkçe Metinlerin Sınıflandırılmasında Metin Temsil Yöntemlerinin Performans Karşılaştırılması/A Comparison of Text Representation Methods for Turkish Text Classification", *EMO Bilimsel Dergi*, Vol. 2, No. 4, 2012.

14. Peng, H., F. Long and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 27.

15. Neviarouskaya, A., H. Prendinger and M. Ishizuka, "Textual affect sensing for sociable and expressive online communication", *Affective Computing and Intelligent Interaction*, pp. 218–229, 2007.

16. Thelwall, M., K. Buckley, G. Paltoglou, D. Cai and A. Kappas, "Sentiment strength detection in short informal text", *Journal of the Association for Information Science and Technology*, Vol. 61, No. 12, pp. 2544–2558, 2010.

17. Witten, I. H., E. Frank, M. A. Hall and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann Publications, 4th edn., 2016, isbn:978-0-12-804291-5.

18. Kaya, M., "Sentiment analysis of Turkish political columns with transfer learning", *Diss, Middle East Technical University*, 2013.

19. Pak, A. and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining.", *LREc*, Vol. 10, 2010.

20. Kökciyan, N., A. Celebi, A. Özgür and S. Üsküdarlı, "Bounce: Sentiment classification in Twitter using rich feature sets", *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Vol. 2, pp. 554–561, 2013.

21. Rosenthal, S., N. Farra and P. Nakov, *SemEval Sentiment Analysis in Twitter*, 2017, `http://alt.qcri.org/semeval2017/task4/`, accessed at January 2017.

22. Erogul, U., "Sentiment analysis in Turkish", *Middle East Technical University, Ms Thesis, Computer Engineering*, 2009.

23. Eryiğit, G., F. S. Cetin, M. Yanık, T. Temel and I. Çiçekli, "Turksent: A sentiment annotation tool for social media", *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 131–134, 2013.

24. Turksent, *Annotation tool developed specifically for manual sentiment analysis of social media posts*, 2010, `http://www.turksent.com/`, accessed at May 2017.

25. Vural, A. G., B. B. Cambazoglu, P. Senkul and Z. O. Tokgoz, "A framework for sentiment analysis in turkish: Application to polarity detection of movie reviews in turkish", *Computer and Information Sciences III*, pp. 437–445, Springer, 2013.

26. Atan, S., "Sentiment analysis by text mining and an application on Borsa Istanbul stock exchange market", *Diss, Ankara University*, 2016.

27. Amanet, H., "Sentiment analysis in turkish social media texts", *Diss, Karadeniz Technical University*, 2017.

28. Turkmen, H., "Discovering product features from Turkish reviews by using aspect based sentiment analysis", *Diss, Kocaeli University*, 2016.

29. Kanburoglu, A. B., "Graph clustering approach to sentiment analysis", *Diss, Isık University*, 2018.

30. Brody, S. and N. Elhadad, "An unsupervised aspect-sentiment model for online reviews", *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 804–812, Association for Computational Linguistics, 2010.

31. Pang, B. and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts", *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, p. 271, Association for Computational Linguistics, 2004.

32. Turney, P. D., "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews", *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 417–424, Association for Computational Linguistics, 2002.

33. Peng, H., *Minimum Redundancy Maximum Relevance Feature Selection (mRMR)*, 2012, `http://home.penglab.com/proj/mRMR/`, accessed at May 2018.

34. Hochreiter, S. and J. Schmidhuber, "Long short-term memory", *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

35. Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

36. Ho, T. K., “Random decision forests”, *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, Vol. 1, pp. 278–282, IEEE, 1995.

37. Breiman, L., “Random forests”, *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.

38. Qi, Y., “Random forest for bioinformatics”, *Ensemble machine learning*, pp. 307–323, Springer, 2012.

39. John, G. H. and P. Langley, “Estimating continuous distributions in Bayesian classifiers”, *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 338–345, Morgan Kaufmann Publishers Inc., 1995.

40. Riloff, E., S. Patwardhan and J. Wiebe, “Feature subsumption for opinion analysis”, *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 440–448, Association for Computational Linguistics, 2006.

41. Bifet, A. and E. Frank, “Sentiment knowledge discovery in twitter streaming data”, *International conference on discovery science*, pp. 1–15, Springer, 2010.

42. Wilson, T., J. Wiebe and R. Hwa, “Recognizing strong and weak opinion clauses”, *Computational Intelligence*, Vol. 22, No. 2, pp. 73–99, 2006.

43. Thelwall, M., K. Buckley and G. Paltoglou, “Sentiment in Twitter events”, *Journal of the Association for Information Science and Technology*, Vol. 62, No. 2, pp. 406–418, 2011.

44. Brooke, J., M. Tofiloski and M. Taboada, “Cross-Linguistic Sentiment Analysis: From English to Spanish.”, *RANLP*, pp. 50–54, 2009.

45. Thelwall, M., K. Buckley and G. Paltoglou, *SentiStrength: Sentiment strength detection in short informal text*, 2010, `http://sentistrength.wlv.ac.uk/`, accessed at May 2017.

46. Harris, Z. S., "Transfer grammar", *International Journal of American Linguistics*, Vol. 20, No. 4, pp. 259–270, 1954.

47. Bengio, Y., R. Ducharme, P. Vincent and C. Jauvin, "A neural probabilistic language model", *Journal of machine learning research*, Vol. 3, No. Feb, pp. 1137–1155, 2003.

48. Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank", *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

49. Blacoe, W. and M. Lapata, "A comparison of vector-based representations for semantic composition", *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pp. 546–556, Association for Computational Linguistics, 2012.

50. Eskenazi, M., G.-A. Levow, H. Meng, G. Parent and D. Suendermann, *Crowdsourcing for speech processing: Applications to data collection, transcription and assessment*, John Wiley & Sons, 2013.

51. Eksisozluk, *Eksi Sozluk: Open Dictionary*, `http://www.eksisozluk.com`, accessed at January 2017.

52. KemikNLP, *Kemik Natural Language Processing Group*, 2014, `http://www.kemik.yildiz.edu.tr/`, accessed at May 2018.

53. Google, *Pre-trained Google news corpus word vector model*, 2014, `https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit`, accessed at May 2018.

54. Mikolov, T., K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space", *arXiv preprint arXiv:1301.3781*, 2013.

55. Deeplearning4j, D. T., *Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0.*, 2017, `http://deeplearning4j.org`, accessed at May 2018.

56. Chollet, F., *Keras: The Python Deep Learning library*, 2011, `https://github.com/fchollet`, accessed at May 2018.

57. Akın, A. A., *Zemberek-NLP : Natural Language Processing tools for Turkish*, 2018, `https://github.com/ahmetaa/zemberek-nlp`, accessed at June 2017.

58. StanfordNLP, D. T., *Stanford Natural Language Processing Group Software*, 2014, `http://nlp.stanford.edu/software/`, accessed at December 2017.

59. Pedersen, T., *University of Minnesota Duluth WordNet Stoplist*, 2017, `http://www.d.umn.edu/ tpederse/Group01/WordNet/wordnet-stoplist.html`, accessed at May 2018.

60. Manning, C. D., C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*, MIT press, 1999.